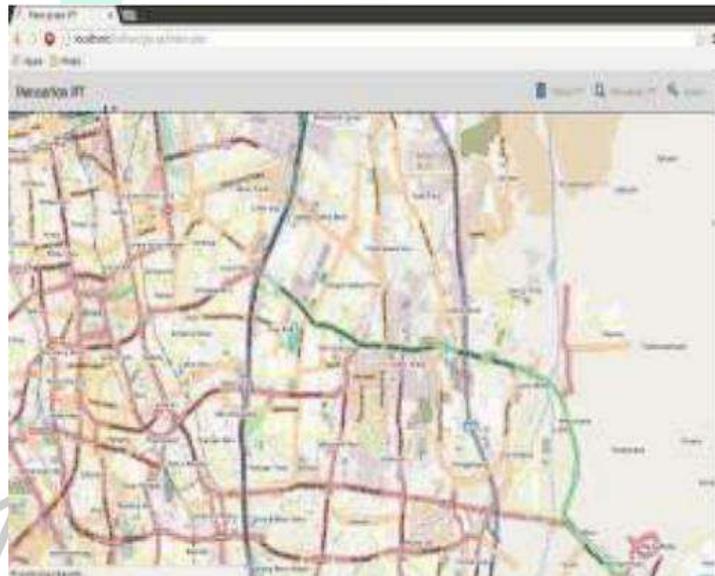


BAB IV PERANCANGAN

Pada bab ini, peneliti melakukan analisis terhadap kebutuhan sistem yang dibuat seperti spesifikasi sistem, cara kerja sistem, rancangan antarmuka, dan rancangan pengujian sistem yang akan digunakan sebagai referensi serta acuan dalam melakukan perancangan aplikasi pemetaan.

4.1 Analisis Sistem Terdahulu

Sistem terdahulu yang dijadikan referensi dalam penelitian ini adalah aplikasi website pencari lokasi perseroan. Pengguna dapat memperoleh informasi berupa nama perusahaan, alamat, no telepon, jenis industry serta jalur menuju lokasi perusahaan yang ada, dan Aplikasi ini hanya memiliki satu jalur yang merupakan hasil pencarian lintasan terpendek.



Gambar 4.1 Antarmuka sistem terdahulu dalam aplikasi pencarian lokasi perseroan

4.2 Spesifikasi Kebutuhan Sistem Baru

Dalam Spesifikasi Kebutuhan Sistem yang baru ini, akan dijelaskan secara rinci bagaimana peneliti memecahkan masalah dalam Spesifikasi Kebutuhan Sistem yang baru. Pada subbab ini terbagi menjadi empat bagian, yaitu diagram konseptual, spesifikasi kebutuhan perangkat lunak, spesifikasi kebutuhan perangkat keras, spesifikasi kebutuhan *Input*, dan spesifikasi kebutuhan *Output*.

4.2.1 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan website terdiri dari fungsionalitas, kinerja, antarmuka pengguna, dan kendala lainnya.

Tabel 4.1 Spesifikasi Perangkat Lunak

No	Perangkat	Keterangan
1	Windows 11 64-bit	Sistem operasi Komputer yang digunakan
2	<i>Visual Studio Code</i>	<i>Platform editor</i> yang digunakan untuk membuat kode program aplikasi
3	<i>PHPMyAdmin</i>	<i>Platform</i> yang digunakan untuk mengelola <i>Database</i>
4	Google Chrome	<i>Platform</i> yang digunakan untuk mengakses <i>web server localhost</i>

4.2.2 Spesifikasi Kebutuhan Perangkat Keras

Spesifikasi komponen fisik yang digunakan untuk menjalankan sistem terdiri dari prosesor, memori, penyimpanan, dan perangkat *Input/Output*.

Tabel 4.2 Tabel Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi yang Digunakan
1	Processor	AMD Ryzen 7 5700U
2	Harddisk	256 GB SSD
3	Memory	8 GB RAM

4.2.3 Spesifikasi Kebutuhan *Input*

Aplikasi pencarian jarak terdekat lokasi *EVCS* pada penelitian ini, tentu diperlukan proses *Input* agar aplikasi dapat berjalan dengan baik. Proses *Input* yang dibutuhkan pada aplikasi ini sebagai berikut.

1. *Input* Lokasi

Untuk menggunakan aplikasi ini, pengguna harus melakukan *Input* lokasi pengguna saat ini, sehingga aplikasi dapat melakukan kalkulasi jarak untuk menuju titik lokasi *EVCS* yang ada.

2. *Input Data User*

Admin dapat menambahkan *User* agar dapat mengelola Aplikasi. *Form* data *User* yang di *Input* pada halaman data *User* berupa nama *User*, nama depan, nama belakang, dan email. *User* yang sudah di *Input* akan ditampilkan pada *list* di halaman *User Data*.

4.2.4 **Spesifikasi Kebutuhan Output**

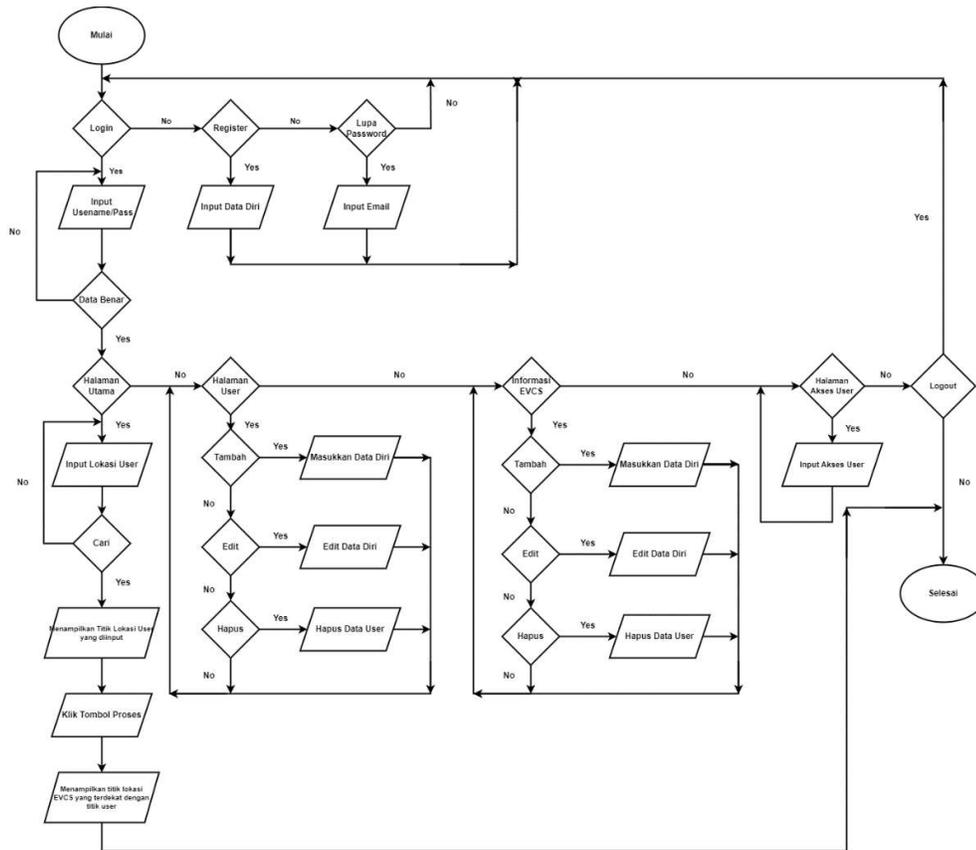
Hasil dari aplikasi pencarian jarak terdekat lokasi *EVCS* berfokus pada jauh garis pada peta dan total jarak. Berikut merupakan spesifikasi *Output* aplikasi.

1. Total jarak setiap titik dari lokasi *Input User*.
2. Menampilkan garis pada setiap titik lokasi *EVCS*
3. Menampilkan rute untuk titik lokasi *EVCS* yang dipilih oleh *User*.
4. *User* yang telah ditambahkan akan ditampilkan pada halaman data *User*.

4.3 **Perancangan Sistem**

Perancangan sistem bertujuan untuk menganalisis kebutuhan sistem agar dapat membangun aplikasi secara menyeluruh. Proses perancangan sistem ini melibatkan penjabaran yang *detail* mengenai prosedur langkah demi langkah, desain, dan perancangan pengujian aplikasi. Berbagai jenis diagram seperti *Flowchart*, *Use Case Diagram*, skenario *Use Case*, *Activity Diagram*, *sequence Diagram* dan desain antarmuka digunakan dalam perancangan sistem untuk mempermudah pemahaman dan visualisasi proses yang terlibat.

4.3.1 Flowchart



Gambar 4.2 Flowchart Sistem Aplikasi Pencarian Jarak Terdekat Lokasi EVCS

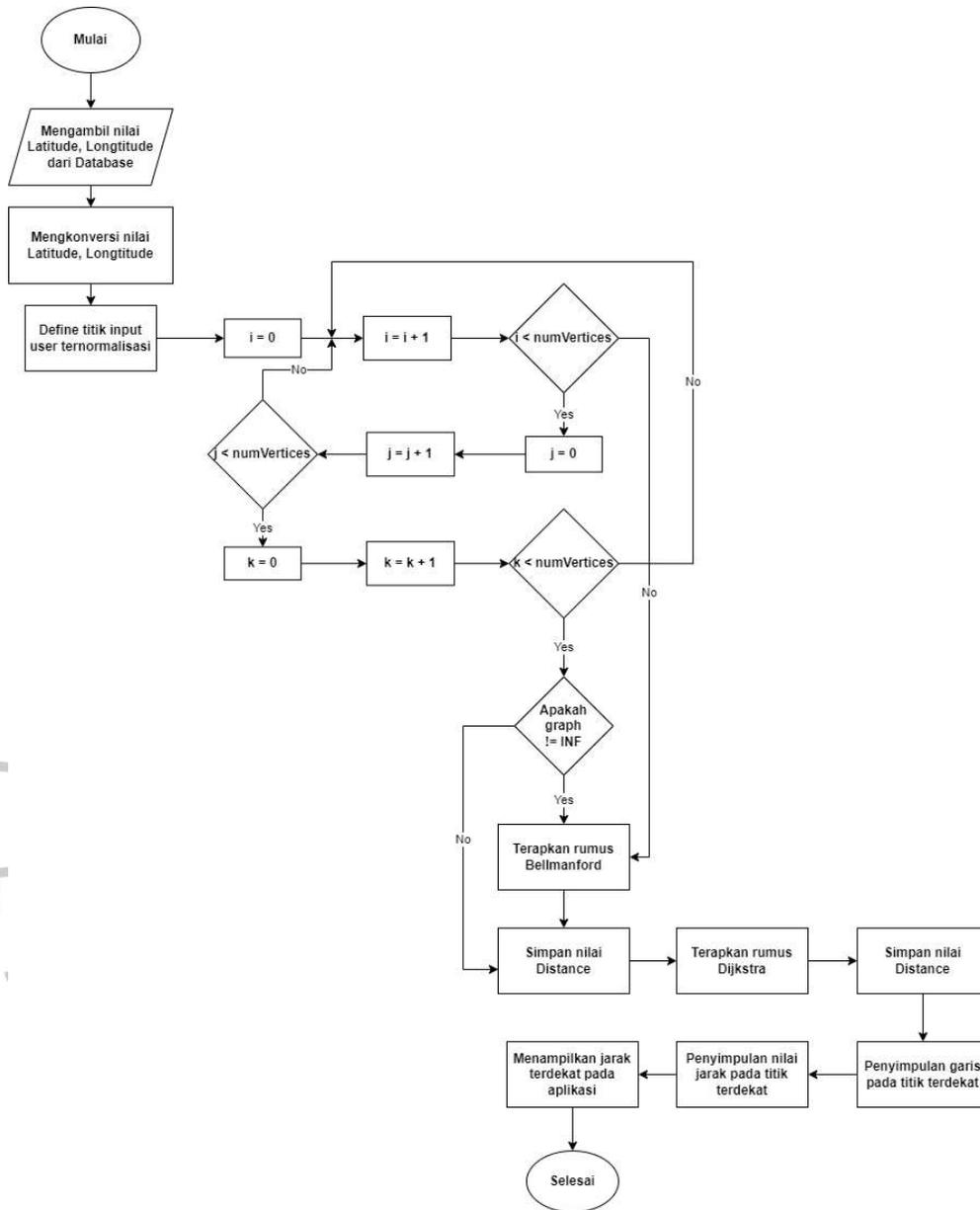
Flowchart adalah suatu diagram yang berguna dalam merepresentasikan urutan langkah-langkah dan keputusan yang dibutuhkan dalam menjalankan suatu proses atau aktivitas dalam sistem tertentu. *Flowchart* memiliki peran penting dalam memvisualisasikan secara jelas dan sistematis alur kerja atau proses yang terjadi dalam suatu sistem atau aktivitas.

Gambar 4.2 diatas merupakan *Flowchart* pada aplikasi *website* pencarian jarak lokasi terdekat EVCS. Penjelasan pada *Flowchart* diatas akan dijelaskan sebagai berikut.

1. *User* dapat mencari titik lokasi EVCS terdekat pada halaman utama. Langkah pertama user memasukkan alamat atau koordinat yang diinginkan *User*, setelah itu sistem akan menampilkan titik lokasi sesuai dengan yang *User* masukkan. Setelah titik awal user telah ditentukan selanjutnya *User*

mengklik tombol proses lalu sistem menampilkan titik lokasi *EVCS* yang terdekat dari lokasi *User*.

2. *User* dapat melakukan *Login* jika telah memiliki account untuk masuk ke website admin. Langkah pertama, admin harus *Input* email dan *Password* dengan benar agar dapat masuk ke dalam website admin. *User* akan langsung disajikan dengan tampilan website admin dengan menu tambahan.
3. Admin dapat mengakses halaman *User Data* yang berisikan data-data *User* yang sudah terdaftar dan dapat mengelola data tersebut seperti melakukan edit, tambah, dan juga menghapus data *User*.
4. Admin dapat mengakses halaman Informasi *EVCS* yang berisikan data-data lokasi *EVCS*, Admin dapat mengelola data tersebut seperti melakukan edit, tambah, dan juga menghapus data *User*.
5. Admin dapat juga mengakses halaman Akses *User*, pada halaman tersebut berisikan beberapa tombol yang dapat memberikan *User* akses lebih terhadap aplikasi website.
6. Setelah selesai mengelola data pada aplikasi website Admin dapat keluar dari website tersebut dengan mengklik tombol *Logout* pada aplikasi, setelah itu Admin akan diarahkan kembali ke halaman utama sebagai *User* biasa.



Gambar 4.3 Flowchart Algoritma Pencarian Jarak Terdekat

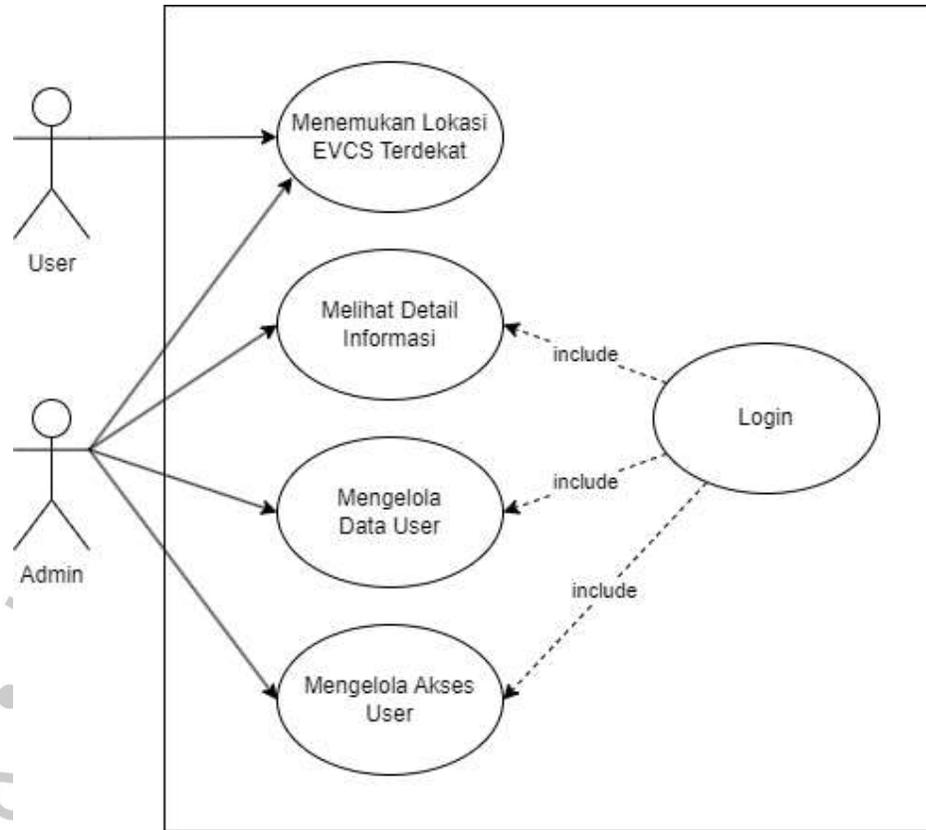
Gambar 4.3 merupakan alur proses dari Algoritma Johnson. Penjelasan pada Flowchart diatas akan dijelaskan sebagai berikut.

1. Mulai,
2. Sistem akan mengambil nilai Longtitude dan Latitude dari setiap titik lokasi EVCS yang sudah didaftarkan pada Database,
3. Lalu nilai dari Longtitude dan Latitude tersebut dikonversikan menjadi nilai jarak dengan menggunakan rumus Haversine,

4. Setelah nilai didapat, sistem akan mendefinisikan nilai titik awal user yang telah dinormalisasi dengan nilai 0,
5. Setelah itu dilakukan pengulangan untuk variable i (lokasi awal) sesuai dengan banyak data dalam database,
6. Lalu selanjutnya dilakukan pengulangan untuk variable j (lokasi tujuan) sesuai dengan banyak data dalam database,
7. Selanjutnya dilakukan pengulangan untuk variable k (nilai jarak) sesuai dengan banyak data dalam database,
8. Kemudian dalam sistem akan dilakukan pengecekan nilai graph ($\neq \text{INF}$), jika nilai tersebut TRUE, maka function Bellmanford akan dieksekusi,
9. Setelah function dari Bellmanford selesai dieksekusi, nilai yang didapat akan disimpan pada variable *Distance*,
10. Lalu nilai pada *Distance* akan diproses Kembali dengan menggunakan
 - function Dijkstra pembobotan ulang nilai *Distance*,
11. Setelah itu simpan kembali nilai yang didapat pada function *Distance*,
12. Langkah akhir yaitu sistem akan melakukan penyimpulan garis dan nilai jarak pada setiap titik lokasi *EVCS*,
13. Setelah itu sistem akan menampilkan garis pada halaman utama menuju titik lokasi *EVCS* terdekat dari lokasi *User*,
14. Selesai.

4.3.2 Use Case Diagram

Use Case Diagram dibawah ini merupakan alur dari proses kerja antara pengguna dengan sistem aplikasi pencarian jarak terdekat lokasi *EVCS*. Berikut adalah penjelasannya.



Gambar 4.4 Use Case Diagram Sistem Aplikasi

Pada Gambar 4.4 menunjukkan bahwa *Actor* dari sistem aplikasi memiliki peran masing-masing. *User* dapat mengakses aplikasi, kemudian *User* dapat melakukan *Input* pada aplikasi untuk lokasi awal *User*, setelah itu sistem akan memproses hasil *Inputan* tersebut, dan menampilkan rute terdekat untuk setiap titik lokasi yang ada. Setelah itu sistem juga akan memunculkan informasi terkait titik lokasi *EVCS* terdekat yang ada, lalu akan menampilkan rute ke titik lokasi tersebut. Sedangkan *Admin* memiliki seluruh akses terhadap aplikasi website, mulai dari melihat keseluruhan detail informasi lokasi titik *EVCS*, mengelola *Data User*, dan mengelola akses *User*.

4.3.2.1 Skenario Use Case Diagram

Skenario *Use Case* adalah untuk menggambarkan dengan jelas bagaimana pengguna atau aktor akan menggunakan sistem dalam situasi nyata guna memastikan bahwa sistem tersebut dapat berfungsi dengan efektif dan efisien. Berikut merupakan penjelasan dari *Use Case Diagram* yang dijelaskan menggunakan tabel skenario.

Tabel 4.3 Skenario *Use Case* Pencarian Jarak Lokasi *EVCS* Terdekat

ID	UC1
Nama <i>Use Case</i>	Menemukan Lokasi <i>EVCS</i> Terdekat
Aktor	<i>User</i> .
Deskripsi	<i>User</i> memasukkan lokasi tertentu.
<i>Pre-Condition</i>	-
Langkah-langkah	<i>User</i> memasukkan lokasi <i>User</i> saat ini pada <i>Search Bar</i> .
<i>Post-Condition</i>	Aplikasi akan menampilkan titik lokasi <i>User</i> , dan menampilkan seluruh lokasi <i>EVCS</i> terdekat.

Tabel 4.3 menunjukkan skenario *Use Case Diagram* untuk pencarian Jarak Lokasi *EVCS* terdekat. Dengan cara aplikasi akan menampilkan titik *User* dan seluruh lokasi *EVCS* terdekat.

Tabel 4.4 Skenario *Use Case Login*

ID	UC3
Nama <i>Use Case</i>	<i>Login</i> .
Aktor	Admin.
Deskripsi	Admin masuk dengan cara <i>Login</i> pada aplikasi dengan <i>Username</i> dan <i>Password</i> yang sudah terdaftar
<i>Pre-Condition</i>	-
Langkah-langkah	<ul style="list-style-type: none"> - Admin memasukkan memasukkan <i>Username</i> dan <i>Password</i> yang sudah terdaftar sebelumnya. - Klik Tombol <i>Login</i>.

<i>Post-Condition</i>	Admin akan dialihkan ke Halaman Utama aplikasi.
-----------------------	---

Tabel 4.4 menunjukkan skenario *Use Case Diagram* untuk melakukan *Login*. *User* diharuskan mengisi *Username* dan *Password* yang telah terdaftar pada *Form Login*.

Tabel 4.5 Skenario *Use Case* Lokasi *User*

ID	UC4
Nama <i>Use Case</i>	Lokasi <i>User</i> .
Aktor	<i>User</i> .
Deskripsi	<i>User</i> memasukkan lokasi tertentu.
<i>Pre-Condition</i>	-
Langkah-langkah	<i>User</i> memasukkan lokasi <i>User</i> saat ini pada <i>Search Bar</i> .
<i>Post-Condition</i>	Aplikasi akan menampilkan titik lokasi <i>User</i> , dan menampilkan seluruh lokasi <i>EVCS</i> terdekat.

Tabel 4.5 menunjukkan skenario *Use Case Diagram* untuk lokasi *User*. *User* memasukkan lokasi *User* saat ini pada *Search Bar*. Lalu aplikasi akan menampilkan titik lokasi *User* tersebut.

Tabel 4.6 Skenario *Use Case* Mengelola Data *User*

ID	UC4
Nama <i>Use Case</i>	Mengelola Data <i>User</i>
Aktor	Admin.
Deskripsi	Admin dapat mengelola Data <i>User</i> .
<i>Pre-Condition</i>	<i>Login</i> dan klik Menu <i>User Data</i> .

Langkah-langkah	<ul style="list-style-type: none"> - Menambah Data <i>User</i> dengan klik tombol Tambah. - Mengedit Data <i>User</i> dengan klik tombol pensil pada kolom aksi. - Menghapus Data <i>User</i> dengan klik ikon tempat sampah pada kolom aksi.
Post-Condition	Berhasil mengelola <i>User</i> Data.

Tabel 4.6 menunjukkan skenario *Use Case* Diagram untuk Kelola *User* Data. Admin dapat melakukan tambah data, mengedit data, dan menghapus data *user* data yang sudah terdaftar.

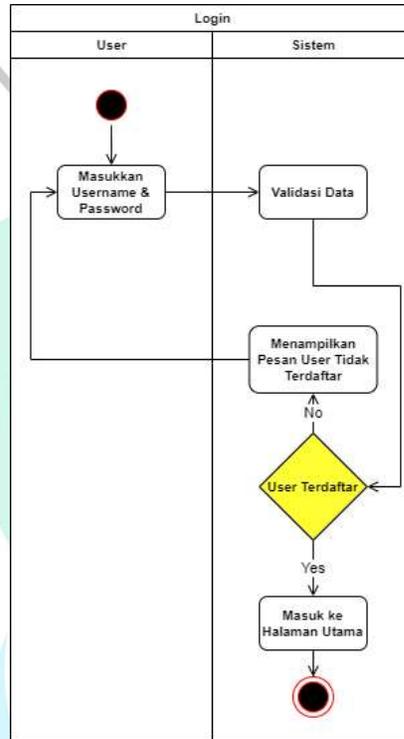
Tabel 4.7 Skenario *Use Case* Kelola Akses *User*

ID	UC6
Nama <i>Use Case</i>	Kelola Akses <i>User</i>
Aktor	Admin.
Deskripsi	Admin dapat mengelola Akses <i>User</i> .
<i>Pre-Condition</i>	<i>Login</i> dan klik Menu Akses <i>User</i> .
Langkah-langkah	Mencentang <i>Option</i> pada <i>Checkbox</i> untuk menghilangkan akses pada <i>Group User</i> tertentu.
<i>Post-Condition</i>	Berhasil mengelola Akses <i>User</i> .

Tabel 4.7 menunjukkan skenario *Use Case* Diagram untuk Kelola Akses *User*. Admin dapat mengubah akses *User* dengan mencentak *Option* pada *Checkbox* yang sudah tersedia.

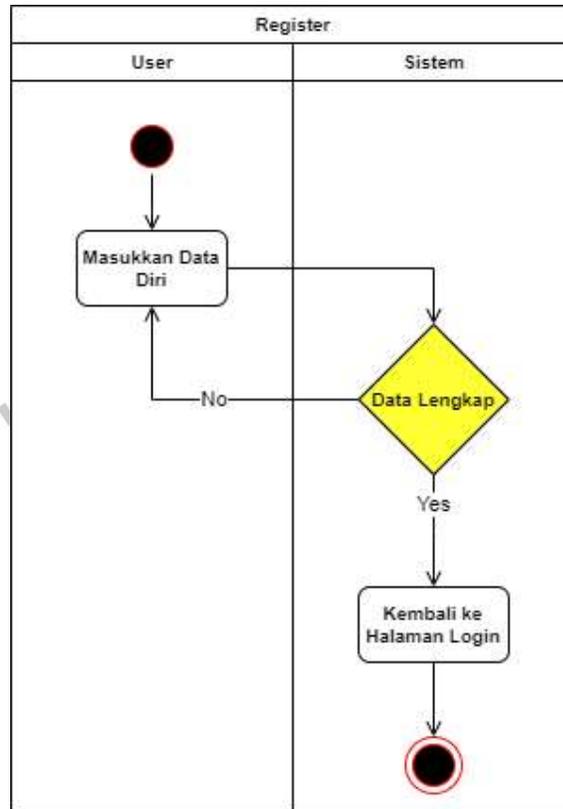
4.3.3 Activity Diagram

Activity Diagram adalah sebuah diagram yang digunakan dalam pengembangan aplikasi untuk menjelaskan urutan dan sistematis dari aktivitas dalam sebuah proses yang telah dijelaskan dalam *Use Case* Diagram sebelumnya. Berikut adalah *Activity* Diagram pada perancangan aplikasi pencarian jarak terdekat lokasi *EVCS*.



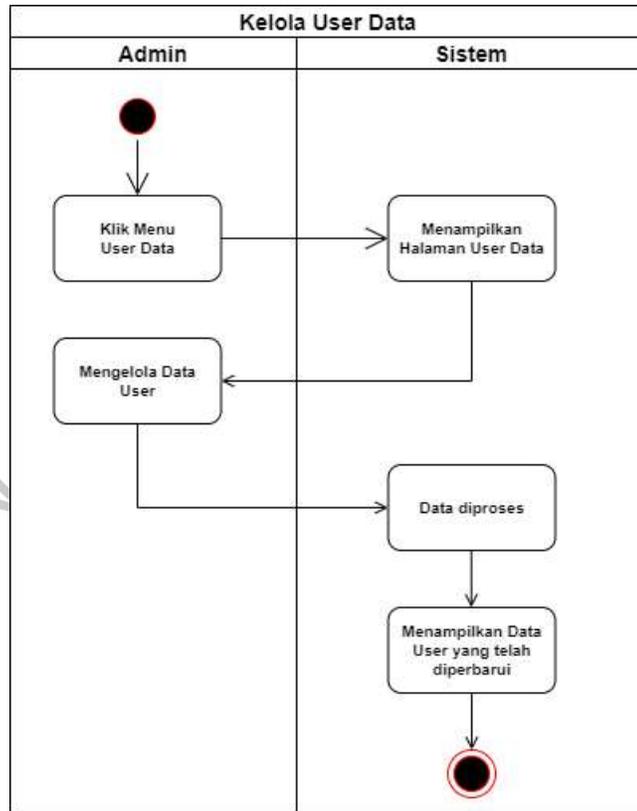
Gambar 4.5 Activity Diagram Login

Gambar 4.5 merupakan *Activity* Diagram untuk fitur *Login*. *User* diharuskan memasukkan *Username & Password* yang telah terdaftar terlebih dahulu. Jika email dan *Password* salah, sistem akan menampilkan peringatan bahwa email atau *Password* salah. Namun jika email dan *Password* benar, maka sistem akan menampilkan pesan bahwa *Login* berhasil dan Admin akan masuk ke Aplikasi halaman *Home*.



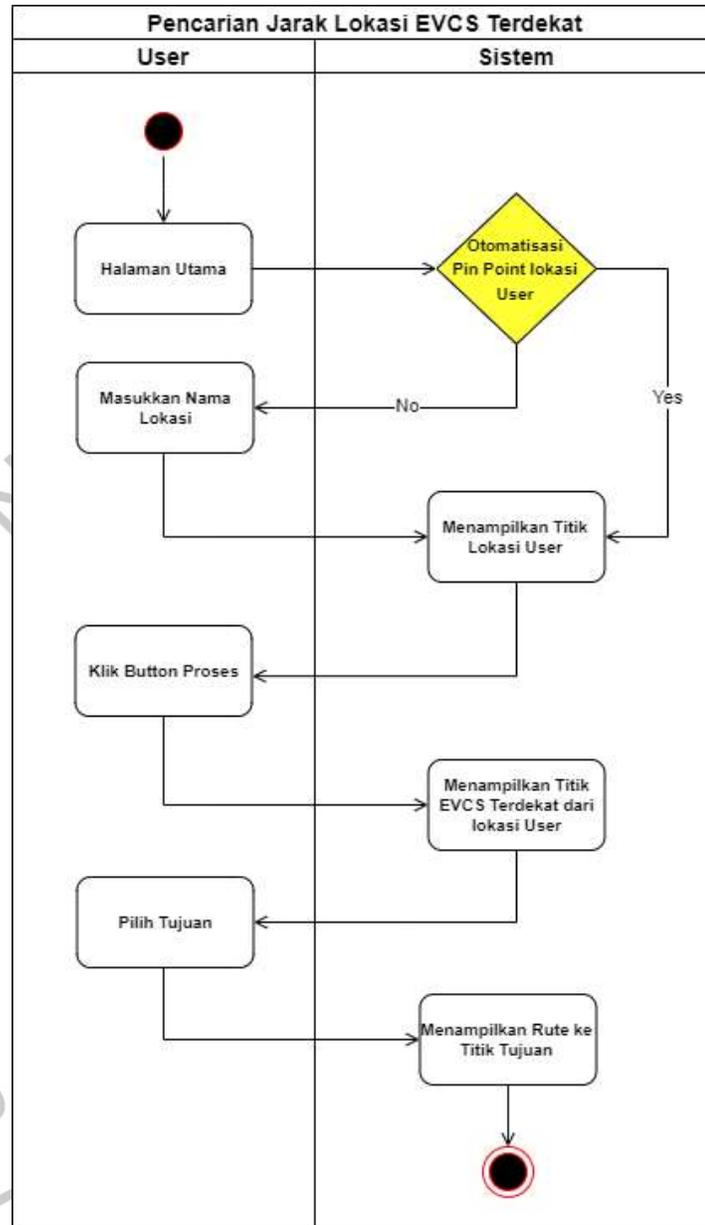
Gambar 4.6 *Activity Diagram Register*

Gambar 4.6 merupakan *Activity Diagram* untuk fitur *Register*. *User* diharuskan memasukkan Data diri untuk mendaftar. Jika data yang *User* masukkan tidak lengkap, sistem akan mengembalikan *User* ke *Form Register*. Namun jika data diri sudah lengkap, maka sistem akan menampilkan pesan bahwa *Register* berhasil dan Admin *User* masuk ke Aplikasi halaman *Home*.



Gambar 4.7 Activity Diagram Kelola User Data

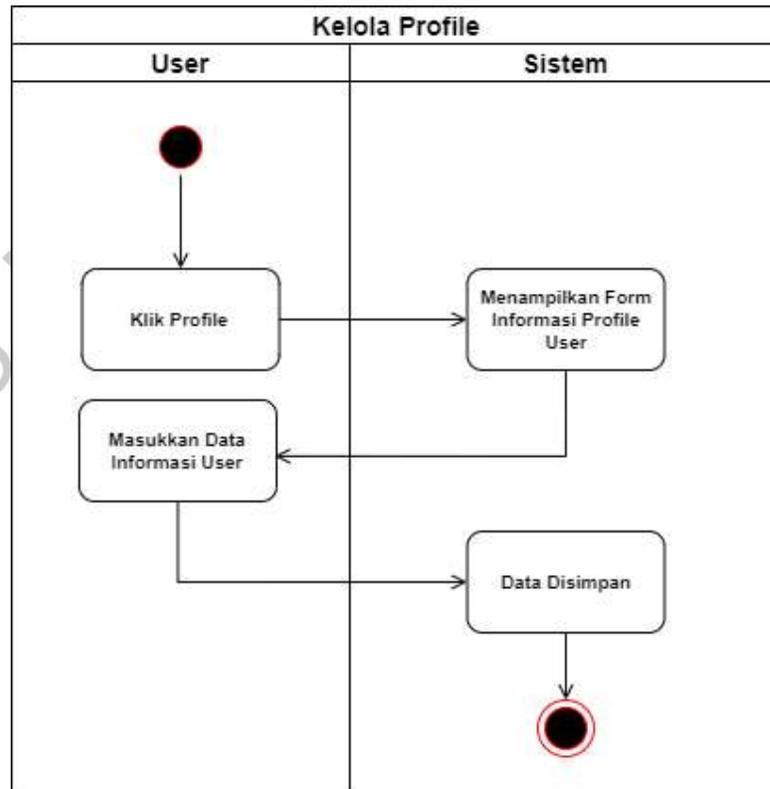
Gambar 4.7 merupakan Activity Diagram kelola User Data. Admin dapat klik menu Data User dan sistem akan menampilkan halaman Data User. Admin dapat menambah User dengan klik button tambah, kemudian mengisi Form tambah. Jika data sudah benar maka data User akan disimpan di dalam database dan ditampilkan dilist data User. Kemudian Admin juga dapat menghapus data User dengan klik ikon tempat sampah pada kolom aksi, lalu data User yang dipilih maka akan terhapus dari tampilan dan dihapus dari database.



Gambar 4.8 Activity Diagram Pencarian Jarak Lokasi EVCS Terdekat

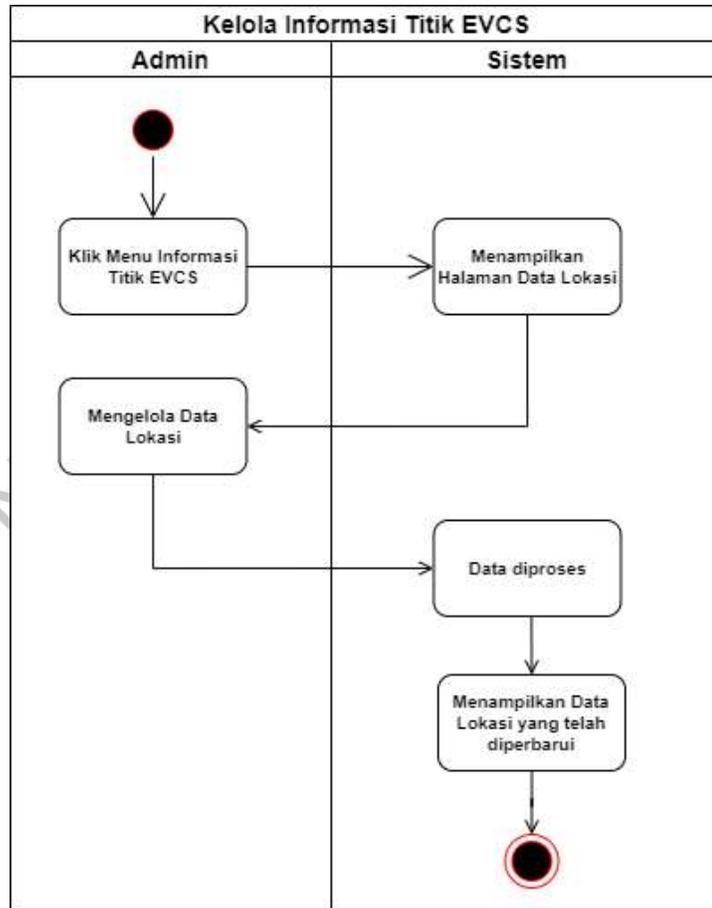
Gambar 4.8 merupakan Activity Diagram Pencarian Jarak Lokasi EVCS. Untuk menggunakan fitur tersebut User terlebih dahulu melakukan Login, lalu Sistem akan membawa User pada Halaman Utama dan setelah itu User diperkenankan untuk memasukkan nama Lokasi yang diinginkan. Lalu klik tombol cari setelah itu Alamat yang dimasukkan akan dieksekusi kedalam Database dan setelah itu sistem akan menampilkan titik lokasi sesuai dengan nama lokasi yang telah User masukkan sebelumnya. Setelah itu klik button proses, sama seperti

Langkah sebelumnya sistem akan memanggil data lokasi yang terdapat pada *database* dan setelah itu sistem akan menampilkan jarak terdekat dari setiap titik lokasi yang ada. Terakhir klik Tujuan maka sistem akan menentukan rute terdekat untuk lokasi yang diinginkan.



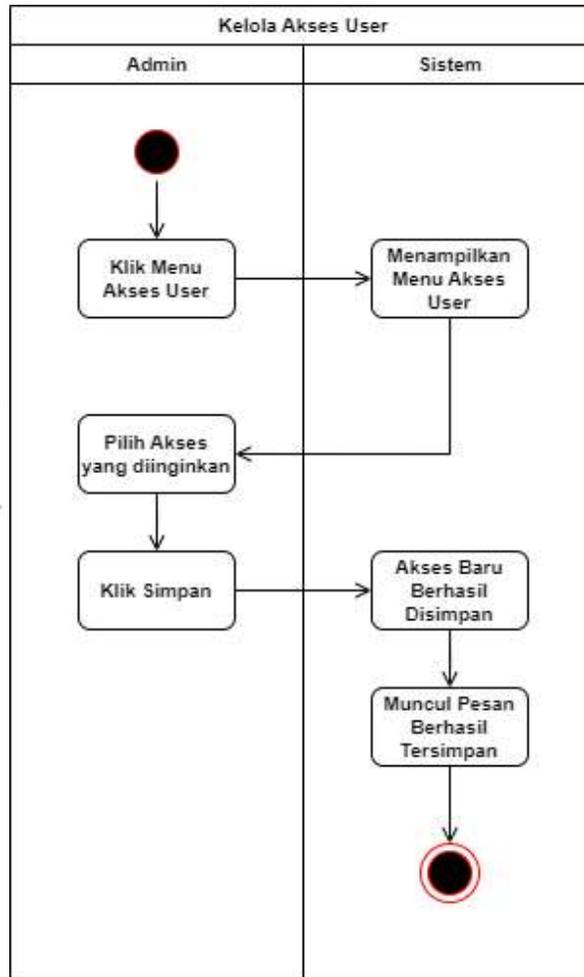
Gambar 4.9 Activity Diagram Kelola Profile

Gambar 4.9 merupakan Activity Diagram untuk Kelola Profile User. User dapat mengakses halaman Profile dengan mengklik Icon Profile, lalu sistem akan menampilkan Form Informasi User. Lalu User dapat mengubah Informasi data diri didalamnya. Lalu klik simpan, setelah itu data yang telah diubah akan disimpan dalam Database. Setelah itu sistem akan menampilkan Pesan bahwa data telah tersimpan. Selanjutnya User dapat juga mengubah gambar profile pada Form Personal, User dapat menambahkan gambar dalam Form tersebut, klik simpan lalu data yang telah diubah akan disimpan dalam Database. Setelah itu sistem akan menampilkan Pesan bahwa data telah tersimpan.



Gambar 4.10 Activity Diagram Kelola Informasi Titik EVCS

Gambar 4.10 merupakan Activity Diagram kelola Informasi Titik EVCS. Admin dapat klik menu titik EVCS dan sistem akan menampilkan halaman Informasi Titik EVCS. Admin dapat menambah User dengan klik button tambah, kemudian mengisi Form tambah. Jika data sudah benar maka data lokasi akan disimpan di dalam database dan ditampilkan dilist data lokasi. Kemudian Admin juga dapat menghapus data User dengan klik ikon tempat sampah pada kolom aksi, lalu data lokasi yang dipilih maka akan terhapus dari tampilan dan dihapus dari database.



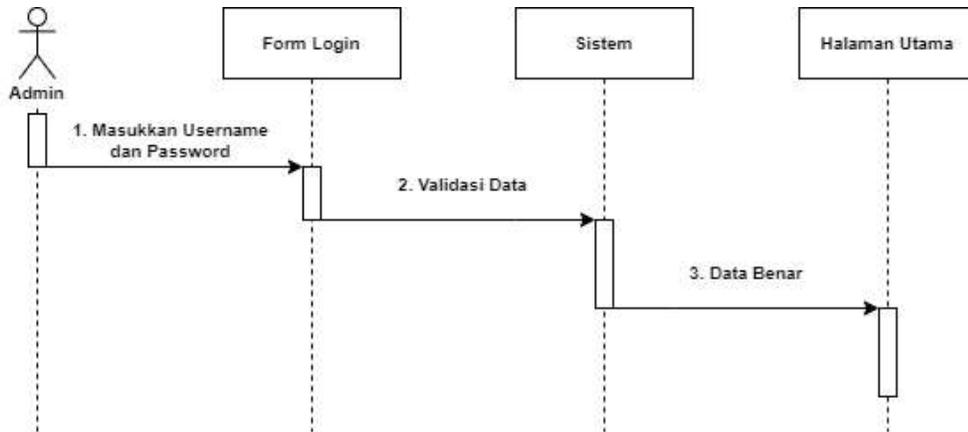
Gambar 4.11 Activity Diagram Kelola Akses User

Gambar 4.11 merupakan Activity Diagram untuk Kelola Akses User. Admin dapat mengklik Menu Akses User, lalu Sistem akan membawa Admin ke Halaman Akses User. Setelah itu Admin dapat mencentang Menu-menu yang dapat diakses oleh User, jika sudah selesai klik Simpan. Lalu Data akses yang sudah dipilih Admin tadi akan disimpan pada Database dan Sistem akan memunculkan pesan bahwa Data Akses berhasil tersimpan.

4.3.4 Sequence Diagram

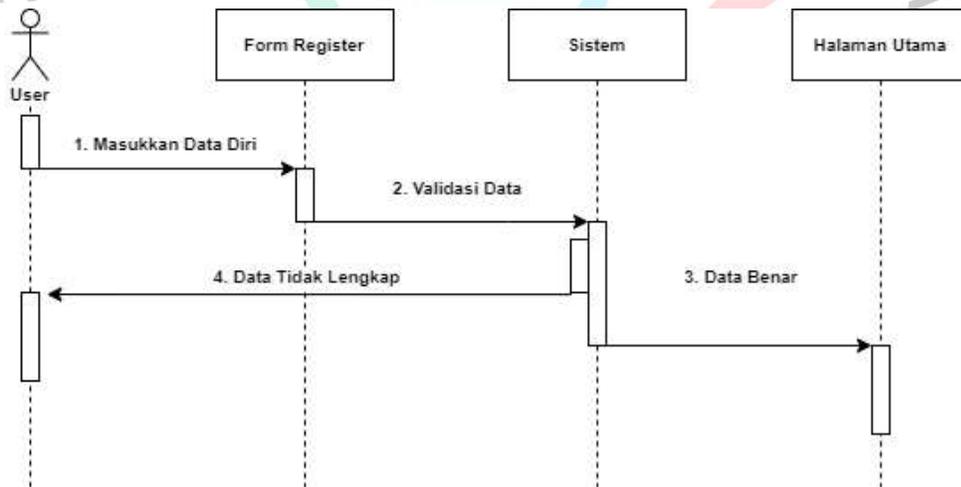
Sequence Diagram digunakan untuk memvisualisasikan urutan pesan atau interaksi yang dilakukan antara objek-objek yang terlibat dalam suatu proses atau kasus penggunaan. Hal ini bertujuan untuk memperlihatkan bagaimana interaksi antara objek-objek tersebut terjadi dan bagaimana keterkaitan antara objek-objek

tersebut dalam suatu sistem. Berikut adalah *sequence* Diagram pada perancangan aplikasi pencarian jarak terdekat lokasi *EVCS*.



Gambar 4.12 *Sequence Diagram Login*

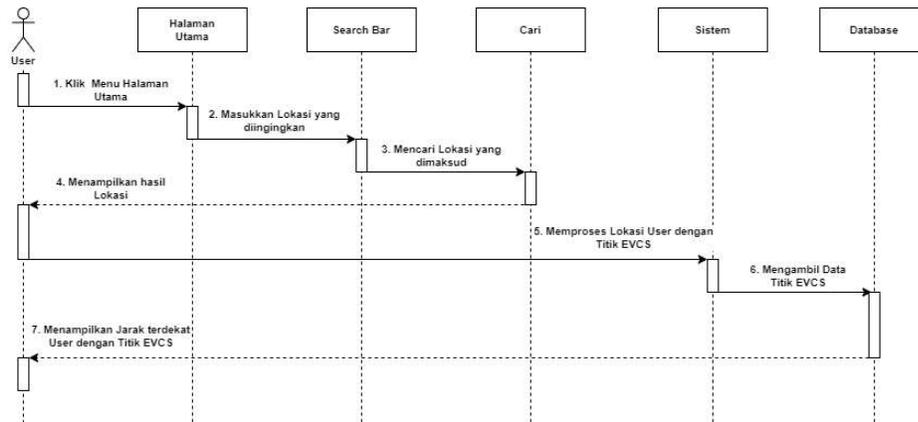
Gambar 4.12 merupakan *Sequence Diagram* yang menjelaskan pengguna yang ingin *Login*. Pengguna memasukkan *Username* dan *Password*. Setelah itu, sistem akan melakukan validasi, jika email dan *Password* salah maka sistem akan menampilkan pesan bahwa email dan *Password* salah. Jika benar maka pengguna akan masuk ke halaman *Home Admin*.



Gambar 4. 13 *Sequence Diagram Register*

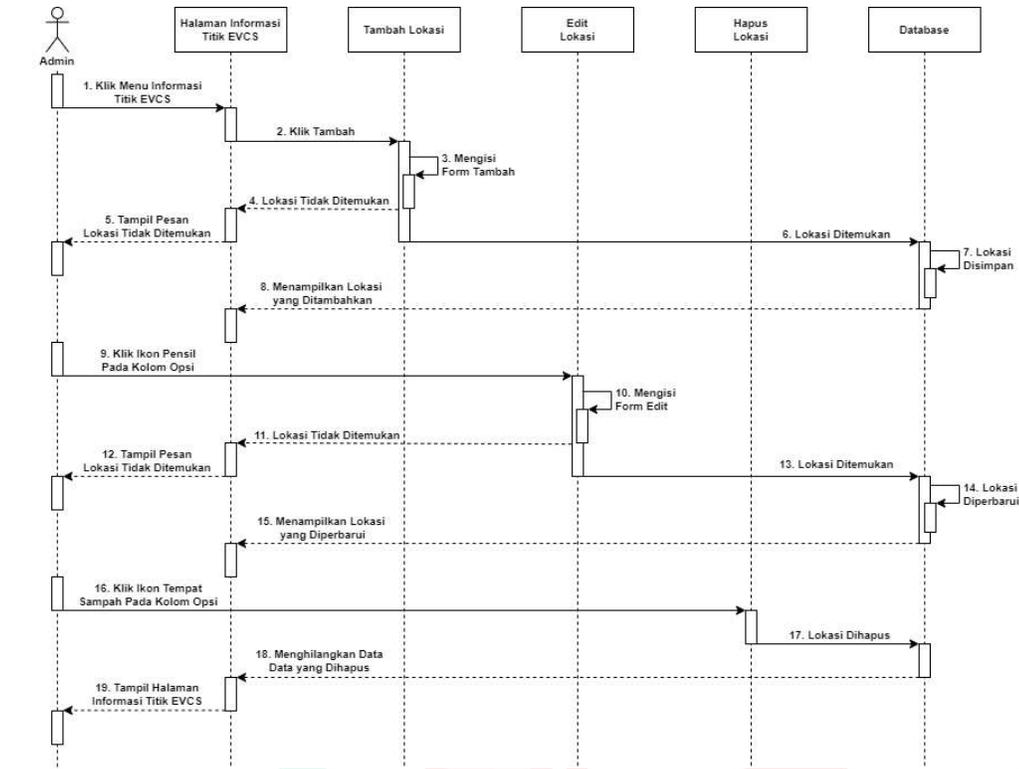
Gambar 4.13 merupakan *Sequence Diagram* yang menjelaskan pengguna yang ingin melakukan registrasi. Pengguna diharuskan memasukkan data diri pada *Form Register* untuk melakukan pendaftaran. Setelah itu, sistem akan melakukan

validasi, apakah data yang dimasukkan oleh *User* sudah lengkap atau belum, jika belum maka sistem akan menampilkan pesan bahwa data diri masih belum lengkap dan mengembalikan *User* ke *Form Register*. Jika benar maka sistem akan menampilkan pesan sukses, dan mengembalikan *User* ke *Form Login*.



Gambar 4.14 *Sequence Diagram* Halaman Utama

Gambar 4.14 merupakan *Sequence Diagram* yang menjelaskan *User* yang ingin mengakses Halaman Utama dan melakukan pencarian Lokasi *EVCS*. *User* mengklik Menu Halaman Utama lalu *User* dapat melakukan *Input* pada *Search Bar*, setelah memasukkan lokasi *User* mengklik tombol cari untuk dapat menampilkan lokasi yang dimaksud, setelah lokasi yang dimaksud sudah muncul pada Halaman Utama Langkah selanjutnya *User* mengklik tombol Proses untuk memulai pencarian jarak terdekat titik lokasi *EVCS*. Titik lokasi *EVCS* didapatkan dari *Database*, setelah titik lokasi *EVCS* sudah diambil sistem akan memunculkan hasil Jarak terdekat *User* dengan Titik *EVCS* yang ada.



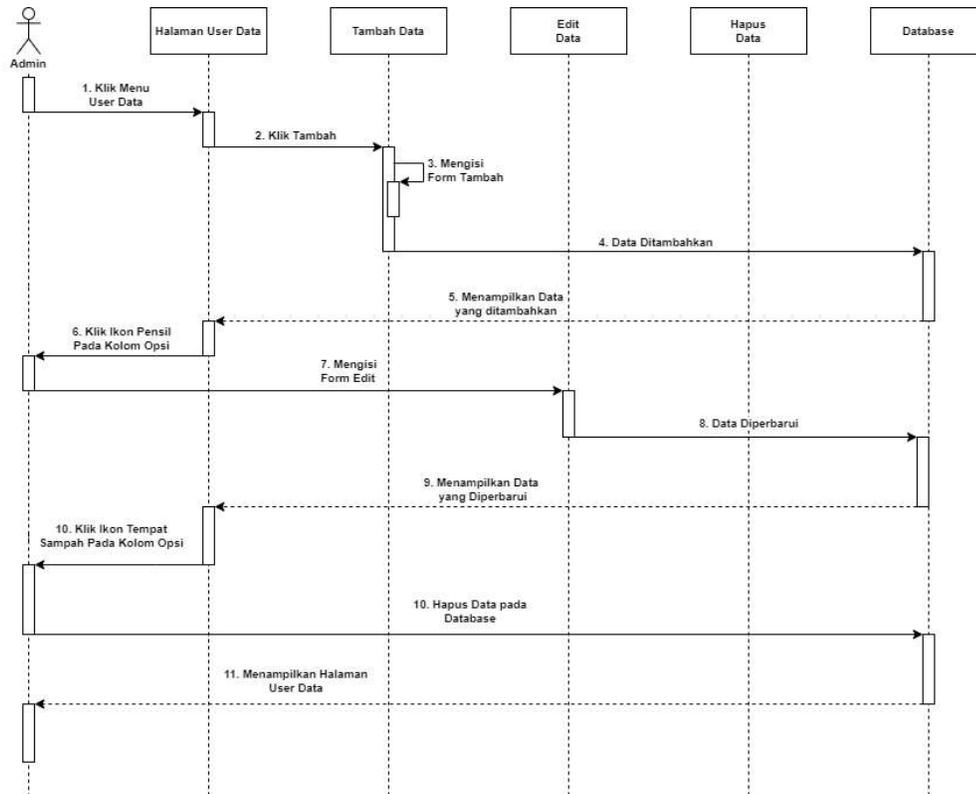
Gambar 4.15 *Sequence* Diagram Halaman Informasi Titik *EVCS*

Gambar 4.15 merupakan *Sequence* Diagram yang menjelaskan pengguna yang ingin kelola Informasi Titik *EVCS*. Pengguna dapat melakukan kelola data lokasi *EVCS* jika sudah melakukan *Login*. Setelah melakukan *Login*, pengguna dapat klik menu Informasi Titik *EVCS* untuk masuk ke halaman Informasi Titik *EVCS*. Berikut merupakan beberapa aktivitas yang dapat dilakukan pengguna untuk mengelola data gambar.

1. Pengguna dapat menambah data lokasi *EVCS* dengan klik tambah. Kemudian mengisi *Form* tambah, apabila terdapat data yang tidak lengkap, sistem akan mengeluarkan pesan yang menunjukkan bahwa data tersebut belum lengkap. Jika data lengkap maka data lokasi yang ditambahkan akan tersimpan di dalam *database* dan sistem akan menampilkan data yang ditambahkan pada *list* halaman data lokasi *EVCS*.
2. Pengguna dapat mengedit data lokasi dengan klik ikon pensil pada kolom opsi. Kemudian mengisi *Form edit*, apabila terdapat data yang tidak lengkap, sistem akan mengeluarkan pesan yang menunjukkan bahwa data tersebut belum lengkap. Jika data lengkap maka data lokasi yang diperbarui

akan tersimpan di dalam *database* dan sistem akan menampilkan data yang diperbarui pada *list* halaman data lokasi *EVCS*.

3. Pengguna dapat menghapus data lokasi dengan klik ikon tempat sampah pada kolom opsi. Kemudian sistem akan menghapus data gambar tersebut dalam *database* dan *list* halaman data lokasi *EVCS*.



Gambar 4.16 *Sequence Diagram* Halaman *User Data*

Gambar 4.16 merupakan *Sequence Diagram* yang menjelaskan pengguna yang ingin kelola *User Data*. Pengguna dapat melakukan kelola data *User* jika sudah melakukan *Login*. Setelah melakukan *Login*, pengguna dapat klik menu *User Data* untuk masuk ke halaman *User Data*. Berikut merupakan beberapa aktivitas yang dapat dilakukan pengguna untuk mengelola data gambar.

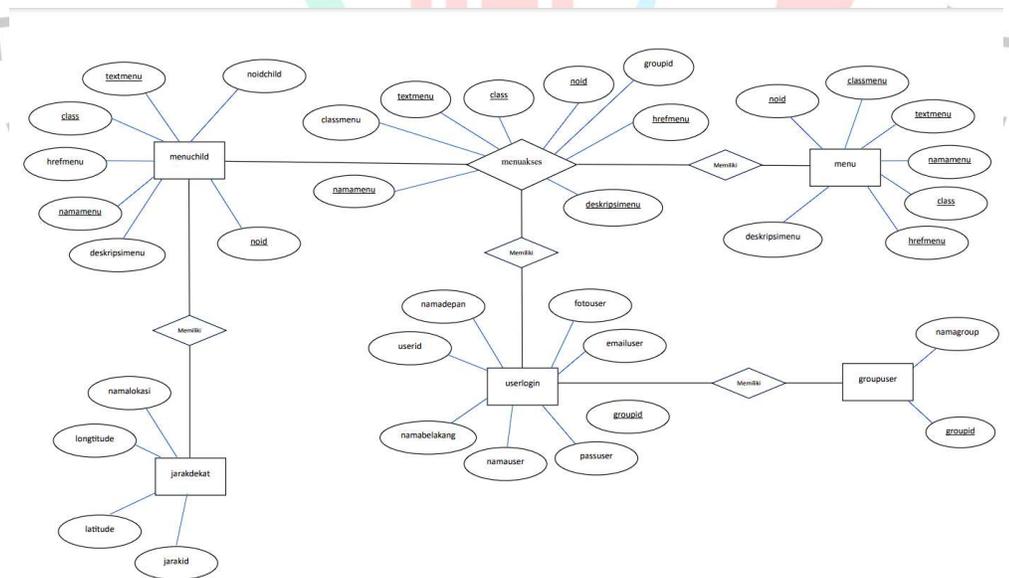
1. Pengguna dapat menambah data *User* dengan klik tambah. Kemudian mengisi *Form* tambah, apabila terdapat data yang tidak lengkap, sistem akan mengeluarkan pesan yang menunjukkan bahwa data tersebut belum lengkap. Jika data lengkap maka data *User* yang ditambahkan akan

tersimpan di dalam *Database* dan sistem akan menampilkan data yang ditambahkan pada *list* halaman data *User*.

2. Pengguna dapat mengedit data *User* dengan klik ikon pensil pada kolom opsi. Kemudian mengisi *Form edit*, apabila terdapat data yang tidak lengkap, sistem akan mengeluarkan pesan yang menunjukkan bahwa data tersebut belum lengkap. Jika data lengkap maka data *User* yang diperbarui akan tersimpan di dalam *Database* dan sistem akan menampilkan data yang diperbarui pada *list* halaman data *User*.
3. Pengguna dapat menghapus data *User* dengan klik ikon tempat sampah pada kolom opsi. Kemudian sistem akan menghapus data gambar tersebut dalam *Database* dan *list* halaman data *User*.

4.3.5 Entity-Relationship Diagram

Entity-Relationship Diagram atau ERD adalah diagram yang digunakan dalam perancangan database untuk menggambarkan entitas, hubungan, dan atribut dalam suatu sistem. Berikut adalah ERD pada aplikasi pencarian jarak terdekat lokasi *EVCS* yang dibuat.



Gambar 4.17 *Entity-Relationship* Diagram Sistem Aplikasi Pencarian Jarak Terdekat Lokasi *EVCS*

4.3.5 Perancangan Basis Data

Perancangan basis data merupakan proses rancangan dari sebuah basis data yang akan digunakan untuk menyimpan data dalam sebuah sistem. Pada aplikasi pencarian jarak terdekat lokasi *EVCS*, peneliti membuat basis data yang diperlukan untuk menyimpan dan mengolah data lokasi *EVCS* dan data *User*. Berikut adalah rancangan basis data pada aplikasi pencarian jarak terdekat lokasi *EVCS*.

Tabel 4.8 Perancangan Basis Data Tabel Group *User*

No	Field	Type	Length	Keterangan
1	<i>groupid</i>	int	3	Id <i>Group</i>
2	<i>namagroup</i>	varchar	20	Nama <i>Group</i>

Tabel diatas merupakan tabel *Group User* yang digunakan untuk menyimpan jenis-jenis *User*, seperti Admin dan *User*.

Tabel 4.9 Perancangan Basis Data Tabel Jarak Dekat

No	Field	Type	Length	Keterangan
1	<i>jarakid</i>	int	3	Id jarak
2	<i>namalokasi</i>	text	-	Nama lokasi
3	<i>latitude</i>	double	-	Nilai latitude
4	<i>longtitude</i>	double	-	Nilai longtitude

Tabel diatas merupakan tabel data jarak yang digunakan untuk menyimpan nama lokasi dan juga koordinat titik *EVCS*.

Tabel 4.10 Perancangan Basis Data Tabel Menu

No	Field	Type	Length	Keterangan
1	<i>noid</i>	int	3	Nomor id
2	<i>class</i>	varchar	15	Kelas data
3	<i>namamenu</i>	varchar	25	Nama menu
4	<i>classmenu</i>	varchar	25	Kelas menu

5	hrefmenu	varchar	70	Koneksi menu
6	textmenu	varchar	25	Teks menu
7	deskripsimenu	varchar	50	Deskripsi menu

Tabel diatas merupakan tabel data menu yang digunakan untuk menyimpan atribut-atribut menu, seperti nama menu, kelas menu, dan deskripsi menu.

Tabel 4.11 Perancangan Basis Data Tabel menuakses

No	Field	Type	Length	Keterangan
1	Groupid	int	3	Group id
2	noid	int	3	Nomor id
3	class	varchar	15	Kelas data
4	namamenu	varchar	25	Nama menu
5	classmenu	varchar	25	Kelas menu
6	hrefmenu	varchar	70	Koneksi menu
7	textmenu	varchar	25	Text menu
8	deskripsimenu	varchar	50	Deskripsi menu

Tabel diatas merupakan tabel menu akses yang digunakan untuk menyimpan atribut-atribut untuk memberikan akses kepada *User*, seperti nama menu, kelas menu, dan deskripsi menu.

Tabel 4.12 Perancangan Basis Data Tabel *User Login*

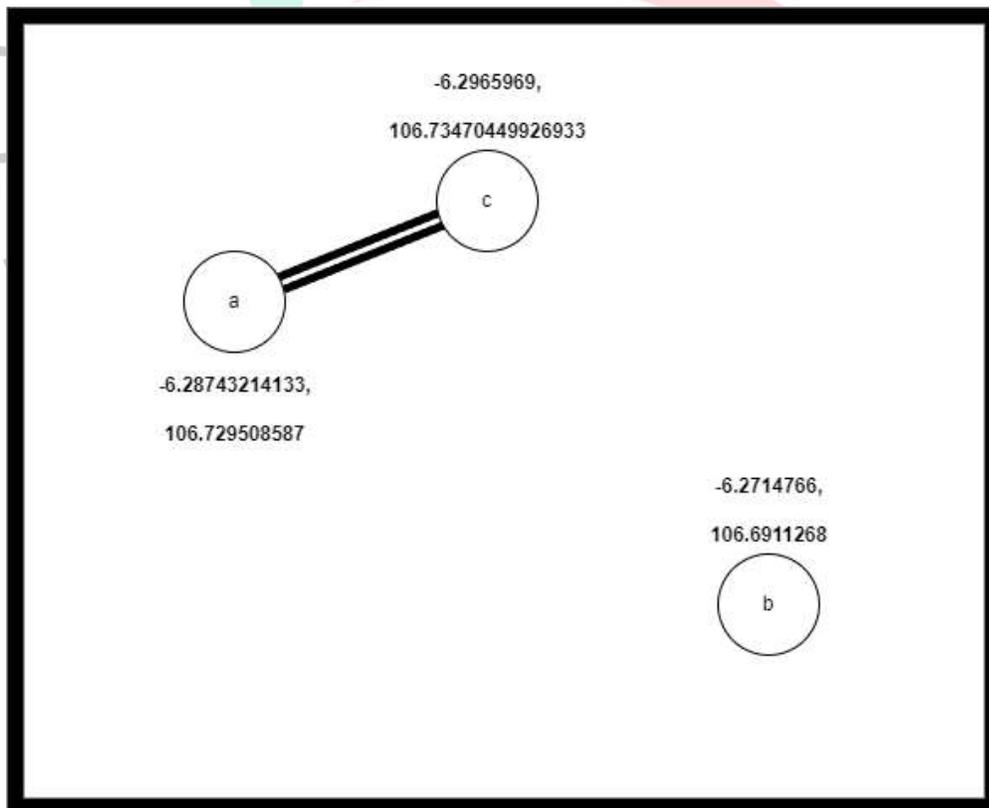
No	Field	Type	Length	Keterangan
1	userid	int	3	Id <i>User</i>
2	namauser	varchar	10	Nama <i>User</i>
3	Passuser	varchar	16	Password <i>User</i>
4	namadepan	varchar	10	Nama depan <i>User</i>

5	namabelakang	varchar	10	Nama belakang <i>User</i>
6	emailuser	varchar	25	Email <i>User</i>
7	fotouser	varchar	1	Foto <i>User</i>
8	groupid	int	3	Group id

Tabel diatas merupakan tabel *User Login* yang digunakan untuk menyimpan data-data yang telah dimasukkan *User* pada halaman register untuk dapat melakukan *Login*.

4.3.6 Perancangan Algoritma

Perancangan algoritma adalah proses merancang langkah-langkah yang sistematis dan terorganisir untuk menyelesaikan suatu masalah atau tugas dengan menggunakan komputasi.



Gambar 4.18 Rancangan *Graph*

Gambar diatas merupakan rancangan dari graph untuk pencarian jarak terdekat dari dua titik lokasi. Terdapat tiga titik yaitu titik a, titik b, dan titik c, masing-masing titik memiliki koordinat yang berbeda. Untuk mengimplementasikan algoritma Johnson pada ketiga titik tersebut maka sebelumnya jarak antar koordinat pada setiap titik harus dihitung terlebih dahulu menggunakan rumus Haversine. Titik a memiliki koordinat -6.28743214133, 106.729508587, dan titik b memiliki koordinat -6.2714766, 106.6911268,

1. Langkah pertama konversikan derajat ke radian,

$$\text{lat1_rad} = -6.28743214133 * (\pi/180)$$

$$\text{lon1_rad} = 106.729508587 * (\pi/180)$$

$$\text{lat2_rad} = -6.2714766 * (\pi/180)$$

$$\text{lon2_rad} = 106.6911268 * (\pi/180)$$

2. Langkah kedua hitung perbedaan lintang dan bujur dengan menambahkan variable baru yaitu dlat, dan dlon,

$$\text{dlat} = \text{lat2_rad} - \text{lat1_rad}$$

$$\text{dlon} = \text{lon2_rad} - \text{lon1_rad}$$

3. Langkah ketiga hitung jarak menggunakan rumus Haversine,

$$a = \sin^2(\text{dlat}/2) + \cos(\text{lat1_rad}) * \cos(\text{lat2_rad}) * \sin^2(\text{dlon}/2)$$

$$c = 2 * \text{atan2}(\text{sqrt}(a), \text{sqrt}(1-a))$$

$$d = r * c$$

Dalam rumus ini, r adalah jari-jari rata-rata bumi, sekitar 6,371 km,

4. Selanjutnya substitusikan nilai yang sudah didapat dan hitung jaraknya,

$$a = \sin^2((-6.2714766 - -6.28743214133) / 2) + \cos(-6.28743214133) * \cos(-6.2714766) * \sin^2((106.6911268 - 106.729508587) / 2)$$

$$c = 2 * \text{atan2}(\text{sqrt}(a), \text{sqrt}(1-a))$$

$$d = 6371 * c$$

Dari perhitungan dengan rumus Haversine diatas didapatkan nilai jarak untuk titik a ke titik b yaitu 3.49 kilometer. Setelah nilai jarak dari titik a ke titik b sudah dapat ulangi langkah-langkah diatas untuk mendapatkan nilai jarak dari titik a ke titik c. Nilai yang didapat untuk titik a ke titik c yaitu 1,19 kilometer.

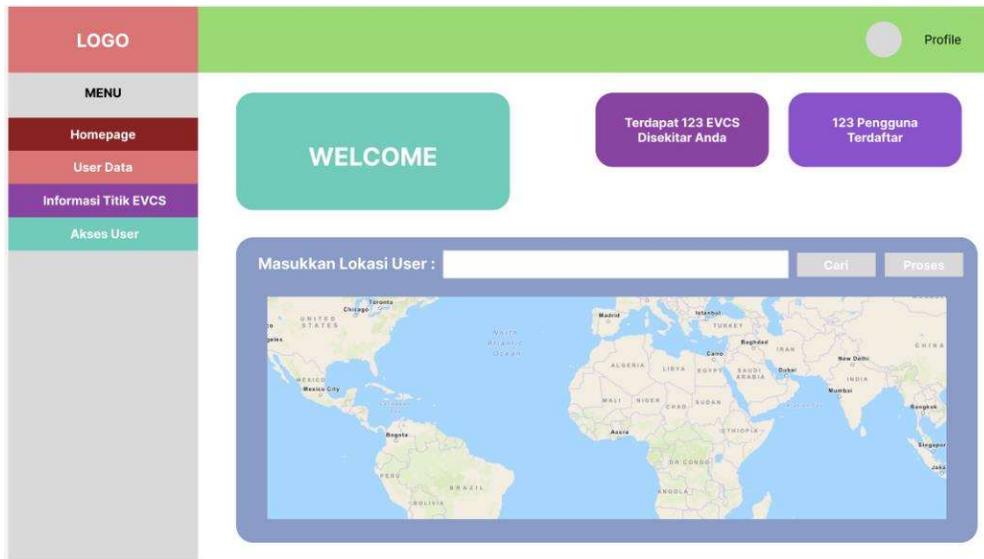
Selanjutnya implementasikan Algoritma Johnson untuk mengetahui jarak terdekat antara titik a ke titik b, dan titik a ke titik c dengan nilai yang sudah dikonversikan.

1. Langkah pertama nilai dari titik a dibandingkan dengan titik b dan c,
2. Langkah kedua nilai jarak pada setiap titik ditimbang ulang menggunakan nilai yang dihitung dengan algoritma Bellman-Ford, Inisialisasi jarak ke semua titik kecuali titik awal dengan tak hingga (∞), dan jarak ke titik awal diatur menjadi 0.
3. Lakukan relaksasi untuk setiap sisi graf (u, v) dengan bobot w , $W(u, v) + H(u) - H(v)$,
4. v adalah jarak terdekat saat ini ke titik v , u adalah jarak terdekat saat ini ke titik u , dan $W(u, v)$ adalah bobot dari sisi yang menghubungkan titik u dan v .
5. Setelah mendapatkan nilai baru gunakan algoritma Dijkstra untuk menemukan jarak terdekat $w : E \rightarrow [0, \infty]$

Setelah langkah-langkah ini selesai, kita akan memiliki jarak terdekat dari titik awal ke semua titik lain dalam graf, serta jalur terdekat dari titik awal ke setiap titik.

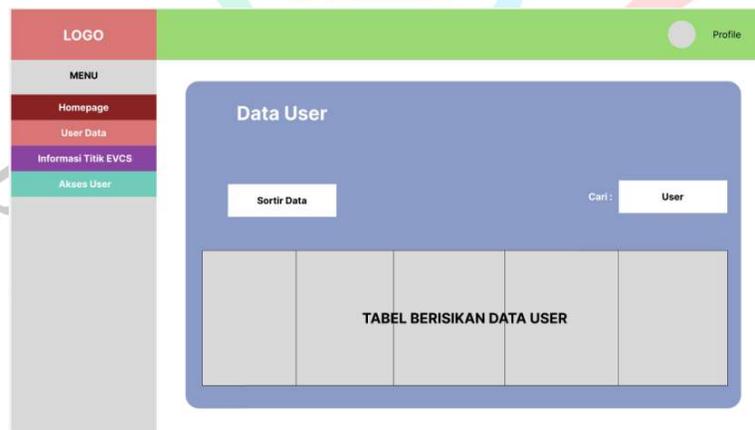
4.3.7 Perancangan Antarmuka

Perancangan antarmuka adalah proses yang bertujuan untuk merancang tampilan visual dan interaksi antara pengguna dan sistem. Selain itu, perancangan antarmuka juga membantu memudahkan programmer dalam mendesain aplikasi yang diinginkan. Berikut merupakan perancangan antarmuka aplikasi pencarian jarak terdekat lokasi *EVCS*.



Gambar 4.19 Perancangan Antarmuka Halaman *Homepage*

Gambar diatas merupakan perancangan antarmuka untuk halaman *Home Admin dan User*. Pada halaman *Home Admin* juga terdapat fitur utama pada aplikasi ini yaitu melakukan pencarian lokasi *EVCS* terdekat dengan memasukkan nama lokasi *User* saat ini terlebih dahulu pada *Search Bar*. Lalu pada halaman tersebut terdapat beberapa menu tambahan seperti *Profile*.



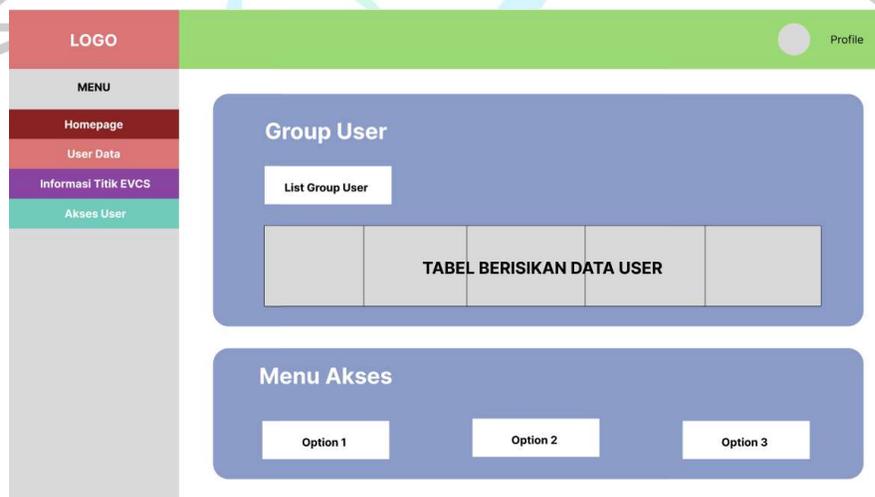
Gambar 4.20 Perancangan Antarmuka Halaman *User Data*

Gambar diatas merupakan perancangan antarmuka untuk halaman *User Data*. Pada halaman *User Data* terdapat fitur seperti penyortiran data *User* dan mencari *User* tertentu. Pada halaman ini juga berisikan Tabel yang berisikan data *User* yang sudah terdaftar.



Gambar 4.21 Perancangan Antarmuka Informasi Titik *EVCS*

Gambar diatas merupakan perancangan antarmuka untuk halaman Informasi Titik *EVCS*. Pada halaman Informasi Titik *EVCS* terdapat fitur seperti penyortiran data lokasi dan mencari lokasi. Pada halaman ini juga berisikan Tabel yang berisikan data lokasi yang sudah terdaftar.



Gambar 4.22 Perancangan Antarmuka Halaman Akses *User*

Gambar diatas merupakan perancangan antarmuka untuk halaman Akses *User*. Pada halaman Akses *User* terdapat fitur untuk Admin memberikan akses

halaman kepada *User*. Pada halaman ini juga berisikan Tabel yang berisikan data *User* yang sudah terdaftar.

4.3.8 Perancangan Pengujian

Perancangan pengujian adalah proses merancang dan mempersiapkan rencana pengujian untuk menguji perangkat lunak atau sistem secara menyeluruh. Perancangan pengujian pada penelitian ini menggunakan pengujian *white box* dan *black box*.

4.3.8.1 Perancangan Pengujian *Black-Box*

Perancangan pengujian *black box* merupakan perancangan pengujian yang diterapkan pada semua fitur sebuah aplikasi untuk menentukan apakah *Output* atau hasil yang diperoleh berkaitan dengan *Input* yang diberikan. Berikut merupakan tabel skenario, *Input*, dan hasil pengujian *black-box*.

Tabel 4.13 Perancangan Pengujian *Black-Box*

No	Skenario Pengujian	Input	Hasil yang Diharapkan
1	<i>Login</i>	Pengguna memasukkan <i>Username</i> dan <i>Password</i> untuk masuk ke aplikasi Admin.	Pengguna dapat <i>Login</i> dan mengakses halaman Utama.
2	<i>Login</i>	Pengguna memasukkan <i>Username</i> dan <i>Password</i> untuk masuk ke aplikasi Admin.	Pengguna tidak berhasil <i>Login</i> dan akan muncul notifikasi yang berisikan bahwa pengguna salah memasukkan <i>Password</i> atau <i>Username</i> .
3	Mencari lokasi <i>User</i>	Pengguna memasukkan lokasi yang diinginkan pada <i>Search Bar</i> .	Pengguna dapat memasukkan lokasi yang diinginkan pada <i>Search Bar</i> .
4	Klik tombol Cari	Pengguna mengklik tombol cari.	Pengguna dapat mengklik tombol Cari.

5	Klik tombol Cari	Pengguna mengklik tombol cari.	Sistem akan memunculkan pesan bahwa alamat atau koordinat yang dimasukkan pengguna tidak dapat ditemukan.
6	Klik tombol Proses	Pengguna mengklik tombol proses.	Pengguna dapat mengklik tombol Proses.
7	Akses Menu <i>User Data</i>	Admin mengklik Menu <i>User Data</i> .	Admin dapat mengklik Menu <i>User Data</i> , dan mengakses halaman <i>User Data</i> .
8	Akses Menu Informasi Titik <i>EVCS</i>	Pengguna mengklik Menu Informasi Titik <i>EVCS</i> .	Pengguna dapat mengklik Menu Informasi Titik <i>EVCS</i> , dan mengakses halaman Informasi Titik <i>EVCS</i> .
9	Akses Menu Akses <i>User</i>	Admin mengklik Menu Akses <i>User</i> .	Admin dapat mengklik Menu Akses <i>User</i> , dan mengakses halaman Akses <i>User</i> .
10	Fitur tambah data untuk menampilkan <i>Form</i> tambah	Admin mengklik fitur tambah data.	Admin dapat mengklik fitur tambah data, dan dapat menambahkan data pada halaman terkait.
11	Fitur ikon pensil untuk menampilkan <i>Form edit</i>	Admin mengklik fitur ikon pensil.	Admin dapat mengklik fitur ikon pensil, dan dapat melakukan <i>edit</i> pada data terkait.
12	Fitur ikon tempat sampah untuk menghapus data	Admin mengklik fitur ikon tempat sampah.	Admin dapat mengklik fitur ikon tempat sampah, dan dapat menghapus pada data terkait.
13	Fitur Menu Akses yang terdapat pada halaman Akses <i>User</i>	Admin men- <i>checklist</i> opsi-opsi akses pada halaman Akses <i>User</i> .	Admin dapat memberikan akses tertentu pada pengguna aplikasi.

14	Akses Profil Menu	Pengguna mengklik Icon Profil.	Pengguna dapat mengakses Profil Menu
15	Simpan data <i>User</i> pada halaman Profil	Pengguna klik tombol simpan	Pengguna dapat menyimpan Informasi data yang telah diubah pada halaman Profil.

4.3.8.2 Perancangan Pengujian *White-Box*

Perancangan pengujian *white-box* merupakan perancangan untuk merancang aplikasi, karena dengan melakukan pengujian ini, dapat memastikan bahwa aplikasi berjalan dengan baik sesuai dengan skenario yang diinginkan.

Tabel 4.14 Perancangan Pengujian *White-Box*

No	Hasil Yang Diharapkan	Source Code	Hasil Pengujian
1	<i>Login</i> dan masuk ke halaman Utama	<pre><div class="col-12 mt-3"> <div class="d-grid"> <button type="button" id="Login" class="btn btn-primary"><i class="fa fa-unlock m-5"></i>Login</button> </div> </div></pre>	
2	Memasukkan lokasi pada <i>Search Bar</i> pada halaman Utama	<pre><div class="col-lg-12 col-xl-12"> <div class="form-group row mb-2"> <label class="col-md-3 col-form- label" for="longitude">Masukkan Alamat atau Lokasi Anda</label> <div class="col-md-7"> <Input class="form-control" id="textsearch" name="textsearch" type="text"> </div></pre>	
3	Tombol Cari	<pre><div class="col-md-2"> <div class="d-grid"> <button class="btn btn-outline-primary" id="search" name="search" type="button"> <i class="fa fa-search m- 5"></i>Cari </button> </div> </div></pre>	

4	Tombol Proses	<pre><div class="col-md-2"> <div class="d-grid"> <button class="btn btn-outline- primary" id="search" name="search" type="button"> <i class="fa fa-search m-5"></i>Cari </button> </div> </div> </div></pre>	
5	Akses Menu User Data	<pre><?php include '../partial/sidebar.php' ?></pre>	
6	Akses Menu Informasi Titik EVCS	<pre><?php include '../partial/sidebar.php' ?></pre>	
7	Akses Menu Akses User	<pre><?php include '../partial/sidebar.php' ?></pre>	
8	Fitur tambah data untuk menampilkan Form tambah	<pre> <button type="button" class="btn btn-pill btn-outline-primary"> <i class='fa fa-table m- 2'></i>Tambah </button> </pre>	
9	Fitur ikon pensil untuk menampilkan Form edit	<pre>"+ "</pre>	
10	Fitur ikon tempat sampah untuk menghapus data	<pre>''+ "";</pre>	
11	Fitur Menu Akses yang	<pre>\$sql = "Insert Into menuakseschild(groupid, noidchild, noid, class, namamenu, hrefmenu, textmenu, deskripsimenu) ";</pre>	

	terdapat pada halaman Akses User	<pre>\$sql .= "Values('\$group', '\$row[0]', '\$row[1]', '\$row[2]', '\$row[3]', '\$row[4]', '\$row[5]', '\$row[6]')"; \$result = MySQL i_query(\$link, \$sql);</pre>	
12	Simpan data User pada halaman Profile	<pre>\$act = \$_GET['act']; if(\$act == 'simpan'){ \$userid = \$_POST['userid']; \$namauser = \$_POST['namauser']; \$namadepan = \$_POST['namadepan']; \$namabelakang = \$_POST['namabelakang']; \$email = \$_POST['emailuser']; \$passuser = md5(\$_POST['namauser'].\$_POST['passuser']); \$groupid = \$_POST['groupid']; \$sql = "Insert Into userLogin(userid, namauser, passuser, namadepan, namabelakang, emailuser, groupid) " . "Values('\$userid', '\$namauser', '\$passuser', '\$namadepan', '\$namabelakang', '\$email', '\$groupid)"; \$result = MySQL i_query(\$link, \$sql); if(\$result){echo 'true';}else{echo 'false';} }</pre>	
13.	Proses konversi nilai Longitude, dan Latitude pada Database	<pre>function Distance(\$lat1, \$lon1, \$lat2, \$lon2, \$unit = 'km') { \$theta = \$lon1 - \$lon2; \$dist = sin(deg2rad(\$lat1)) * sin(deg2rad(\$lat2)) + cos(deg2rad(\$lat1)) * cos(deg2rad(\$lat2)) * cos(deg2rad(\$theta)); \$dist = acos(\$dist); \$dist = rad2deg(\$dist); \$miles = \$dist * 60 * 1.1515; \$unit = strtoupper(\$unit); if (\$unit == "KM") { return (\$miles * 1.609344); } else if (\$unit == "MI") {</pre>	

		<pre> return \$miles; } else { return \$miles * 0.8684; } }</pre>	
--	--	--	--

